

We Claim:

- Sub ar
1. A multi-attribute valuing system comprising:
 - a buyer attribute-input for receiving a plurality of delta values for a product, each delta value indicating a higher or a lower true-value that the buyer places on the product when modified by the attribute associated with the delta value;
 - a seller attribute-input for receiving a plurality of delta values for the product, each delta value indicating a higher or a lower true-value that the seller places on the product when modified by the attribute associated with the delta value;
 - an attribute manager, receiving the attributes and delta values from the buyer and from the seller, for creating and managing attribute trees, each attribute tree specifying dependencies among the attributes of the product, the attributes associated with delta values that adjust a true-value of the product;
 - an attribute-tree analyzer, receiving an attribute tree for the buyer and an attribute tree from the seller for the product, for comparing the delta values from the buyer with the delta values from the seller for a plurality of attributes that modify the product, the attribute-tree analyzer selecting an optimal specification of the product, the optimal specification including a series of the attributes; and
 - a product specifier, coupled to the attribute-tree analyzer, for reporting to the seller the series of attributes for the optimal specification of the product,
 whereby the optimal specification of the product is determined by analyzing delta values specified by the buyer and the seller that adjust the true-value of the product.
 2. The multi-attribute valuing system of claim 1 wherein each branch of the attribute tree specifies an attribute, including an attribute name, an attribute value, and a delta value, whereby the attribute value specifies a value of the attribute that modifies the product itself, while the delta value specifies a true-value change offered to trade the product.

3. The multi-attribute valuing system of claim 2 wherein each branch of the attribute tree also specifies one or more ranges where each range has one or more interval of attribute values.

5 4. The multi-attribute valuing system of claim 2 wherein each attribute tree includes a root, the root for specifying a base product unmodified by the attributes;
wherein a path from the root to a leaf of the attribute tree specifies a final product,
wherein the final product is the base product modified by attributes along the
10 branches in the path from the root to the leaf,
whereby the path through the attribute tree specifies attributes for the final product.

15 5. The multi-attribute valuing system of claim 4 wherein the root of the attribute tree includes a baseline value, the baseline value being a value for the base product;
wherein each node in each branch of the attribute tree includes the delta value, the delta value being a value adjustment for the product modified by the attribute specified by the branch;
wherein a final true-value of the final product is generated by adding the baseline value
20 to a sum of all delta values along a path in the attribute tree from the root to a leaf for the final product,
whereby the final true-value is determined by summing the baseline value with the delta values along a path in the attribute tree.

25 6. The multi-attribute valuing system of claim 2 wherein the product is an item for sale or a service.

7. The multi-attribute valuing system of claim 2 wherein the attributes and the delta values are explicitly defined by the buyer or seller.

8. The multi-attribute valuing system of claim 2 wherein the attributes and delta value are implicit, the delta values being determined from other data.

9. The multi-attribute valuing system of claim 8 further comprising:

5 a data gatherer, coupled to a seller web site, for reading other data from the seller web site, the other data including price data for products with various attributes;
a data store, coupled to the data gatherer, for storing the other data;
an implicit tree builder, coupled to read the other data in the data store, for generating
the attribute tree for the seller web site by comparing the price data in the other
10 data for products with different attributes,
whereby the attribute tree is generated implicitly.

10. The multi-attribute valuing system of claim 9 wherein the data store includes records that contain a product specifier, attributes for the product, and a value for the product with the attributes stored in the record.

11. The multi-attribute valuing system of claim 10 wherein the data store is a relational database.

12. The multi-attribute valuing system of claim 9 further comprising:
heuristic means for re-balancing the attribute tree generated by the implicit tree builder
to produce a sparse representation of the attribute tree,
whereby the attribute tree is re-balanced.

13. The multi-attribute valuing system of claim 12 further comprising:
optimizing means, coupled to the attribute tree, for re-arranging attributes in the
attribute tree, wherein attributes with greater variances in attribute values are
moved toward the root while attributes with lower variances are moved toward
the leaf,

whereby high-variance attributes are moved near the root of the attribute tree during optimization.

14. A computer-program product comprising:

5 a computer-usable medium having computer-readable program code means embodied therein for indicating differential values of products in a product family, the computer-readable program code means in the computer-program product comprising:

10 attribute means for storing product-attributes that specify products within the product family, the products varying according to product-attributes;

input means for receiving delta values from a user, the delta values representing differences in values of products in the product family, the products varying according to the product-attributes, the delta values representing differences in value perceived by the user for the product specified by the product-attributes; and

5 value attach means, coupled to the input means and to the attribute means, for attaching the delta values to the product-attributes to form a user attribute-store, whereby the user attribute-store indicates differences in values perceived by the user for products in the product family that vary in specification according to the product-attributes.

15. The computer-program product of claim 14 wherein the product-attributes include dependencies wherein a value of one product-attribute depends on a value of a second product-attribute, the attribute means including dependency means for storing the dependencies of the product-attributes for the product family,

25 whereby product-attribute dependencies are stored.

16. The computer-program product of claim 15 wherein the user attribute-store comprises an attribute tree having a tree-like logical structure, the attribute tree

30

having a root that specifies a base product in the product family, the root branching into nodes according to the product-attributes, the final terminal nodes being leaves, the leaves each representing a product in the product family, whereby the user attribute store is an attribute tree.

5

17. The computer-program product of claim 16 wherein a differential value of a product is determined by summing delta values for each node along a path from the root to the leaf for the product, the final value of the product being the differential value added to a base value for the base product,

10

whereby product values are determined by summing delta values along a path in the attribute tree.

18. The computer-program product of claim 14 wherein users include buyers and sellers of a product in a product family;

15

wherein a buyer is associated with a buyer attribute-store that is a user attribute-store that stores the delta values perceived by the buyer;

wherein a seller is associated with a seller attribute-store that is a user attribute-store that stores the delta values perceived by the seller;

whereby buyer and seller delta values are stored by user attribute-stores.

20

19. The computer-program product of claim 18 wherein the users also include facilitators and market makers, wherein trading partners include buyers, sellers, facilitators and market makers.

25

20. The computer-program product of claim 18 wherein the computer-readable program code means further comprises:

a value comparator, receiving the buyer attribute-store and the seller attribute-store, for comparing the delta values from the buyer attribute-store and the seller attribute-store for each product-attribute to find an optimal group of product-attributes that specify an optimal product for sale by the seller to the buyer,

30

whereby an optimal product in the product family for a sale is determined by comparing attribute stores of delta values for the buyer and for the seller.

21. The computer-program product of claim 20 wherein the buyer and seller are connected by a network;

wherein the computer-readable program code means further comprises:

network connection means for transmitting and receiving packets from remote clients for the buyer and the seller,

whereby an electronic exchange connects the buyer and the seller.

22. The computer-program product of claim 14 wherein the input means includes an explicit input to receive delta values from the user, whereby the delta values are directly input by the user.

23. The computer-program product of claim 14 wherein the input means includes an implicit input that reads full-values for products in the family, the input means including compare means for determining the delta values for the product-attributes that specify the products in the product family, whereby the delta values are implicitly input from the user.

24. The computer-program product of claim 23 wherein the input means includes scan means for scanning a product and value list to implicitly input the delta values.

25. A computer-implemented method for selecting a product for sale to a trading partner comprising:

receiving a selection of a base product from the trading partner;

receiving a list of attributes for the base product from the trading partner;

for each attribute in the list of attributes, receiving a delta value from the trading partner for the attribute, the delta value indicating an additional value the

trading partner places on the product when the base product is modified by the attribute, the delta values being trading partner-specified;
 selecting as a selected product the product modified by a subset of the attributes in the list of attributes, the subset of the attributes including attributes wherein a sum
 5 of the delta values for attributes in the subset of the attributes is a selected sum;
 offering the selected product to the trading partner for sale to the trading partner;
 whereby the product sold is selected using a selected sum of the delta values of the attributes specified by the trading partner.

10 26. The computer-implemented method of claim 25 further comprising:
 receiving a baseline value from the trading partner, the baseline value indicating a value
 the trading partner places on the base product when not modified by any of the attributes;
 wherein the baseline value from the trading partner is an amount the trading partner is
 15 willing to pay for the base product, and the delta values are additional amounts
 the trading partner is willing to pay for the product modified by the attributes,
 whereby attributes of the selected product for sale to the trading partner are selected
 based on value specified by the trading partner for the attributes.

20 27. The computer-implemented method of claim 26 further comprising:
 dynamically generating a true-value for the selected product based on the selected sum
 of the delta values and the baseline value,
 whereby a sales price is adjusted for trading partner-specified delta values of attributes
 for the selected product.

25 28. The computer-implemented method of claim 25 further comprising:
 creating a trading partner attribute tree from the list of attributes and the delta values
 received from the trading partner, the attribute tree specifying values to the
 trading partner of the product modified by the attributes,

whereby the trading partner specifies values of the product modified by different attributes.

29. The computer-implemented method of claim 28 further comprising:
5 making a connection to the trading partner over the Internet.

30. A computer-implemented tool for finding an optimally-specified product for purchase comprising:

10 a buyer attribute-tree generator for generating a buyer attribute tree, the buyer attribute tree having a base product at a root of the buyer attribute tree, wherein each branch in the buyer attribute tree specifies an attribute that modifies the base product;

base-product means for searching for sellers that are offering the base product for sale, the base-product means generating a sellers list of sellers that offer the base product for sale;

15 a seller attribute-tree generator for generating a plurality of seller attribute trees, each of the seller attribute trees having the base product at a root of the seller attribute tree, wherein each branch in the seller attribute tree specifies an attribute that modifies the base product;

20 an attribute tree comparator, receiving the buyer attribute tree and one of the seller attribute trees, for comparing the buyer attribute tree to the seller attribute tree and determining an overlap, the overlap indicating a degree of overlap of the attributes for the base product;

25 wherein the attribute tree comparator compares seller attribute trees to the buyer attribute tree for all sellers in the sellers list; and

an overlap maximizer, coupled to the attribute tree comparator, for finding a maximum overlap generated by the attribute tree comparator, the maximum overlap being generated for a maximum seller attribute tree of a chosen seller;

30 wherein the chosen seller offers for sale the base product with attributes most-closely matching attributes specified by the buyer in the buyer attribute tree,

whereby the seller is chosen based on maximum attribute overlap by comparing attribute trees.

31. The computer-implemented tool of claim 30 further comprising:

5 a linker, responsive to the chosen seller from the overlap maximizer, for connecting the buyer with the chosen seller to allow the buyer to purchase the base product with the attributes most-closely matching buyer-specified attributes.

32. The computer-implemented tool of claim 31 further comprising:

10 a bid generator, coupled to the chosen seller, for sending a bid from the buyer to the chosen seller, the bid including a specification of the base product and the attributes that overlap the buyer attribute tree and the seller attribute tree for the chosen seller.

15 33. A method for dynamically augmenting a product with attributes that are most valued by prospective buyers comprising:

receiving a plurality of attributes and attribute values for the attributes from the prospective buyers;

for each attribute, and for each prospective buyer,

20 determining when a buyer has submitted an attribute value for the attribute;

when a buyer has submitted the attribute value, adjusting a counter for the attribute value;

examining the counter for each attribute value to find maximum attribute values, the maximum attribute values having the most prospective buyers submitting the attribute value to indicate buyer preference of the attribute with the attribute value;

25 modifying the product according to the attributes with the maximum attribute values to form an augmented product that is more valued by the prospective buyers; and offering the augmented product for sale to the prospective buyers,

augmented with attributes

im 33 wherein adjusting

menting the counter for e

,

cremented for each pro

5

10

$$\begin{array}{ccccccc} \text{H} & \text{H} & \text{H} & \text{H} & \text{H} & \text{H} & \text{H} \\ | & | & | & | & | & | & | \\ \text{C}_1 & \text{C}_2 & \text{C}_3 & \text{C}_4 & \text{C}_5 & \text{C}_6 & \text{C}_7 \\ | & | & | & | & | & | & | \\ \text{H} & \text{H} & \text{H} & \text{H} & \text{H} & \text{H} & \text{H} \end{array}$$